

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-implemented method for controlling access to a data object stored in a ~~non-volatile memory~~ first storage location, the data object having an identifier (ID), the method comprising:

~~checking, before accessing the data object, whether the ID is contained~~ stored in a first lock object and whether the ID is associated with a second storage location; and granting access to ~~accessing~~ the data object[[,]] if the ID is not ~~contained~~ stored in the first lock object ~~or if~~ and the ID is not yet associated with a second storage location;

~~determining whether a transactional lock has been successfully set on the data object;~~

~~determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and~~

~~granting read/write access to the data object based on the permanent lock not being set on the data object.~~

2. (Currently Amended) The method of claim [[6]] 1, further comprising:
deleting the ID from the second first lock object, if the ID is stored in the first lock object and the ID is not yet associated with a second storage location, after granting access to the data object.

3. (Currently Amended) The method of claim 1, wherein the first lock object comprises a table[[,]] having a first column for the ID and a second column for a link to the second storage location associated with the ID.

4. (Currently Amended) The method of claim 2, wherein the first lock object comprises a table[[,]] having a first column for the ID and a second column for a link to the second storage location associated with the ID.

5. (Original) The method of claim 1, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

6. (Currently Amended) The method of claim 1, further comprising:
~~before performing the check,~~ storing the ID in a second lock object before
checking whether the ID is stored in a first lock object and whether the ID is associated
with a second storage location, ~~which is stored in a volatile storage means.~~

7. (Currently Amended) The method of claim 6, further comprising:
checking[[,]] whether the ID ~~has been successfully~~ is stored in the second lock
object before granting access to ~~accessing~~ the data object; and[[,]]
denying access to the data object if the ID ~~has~~ is not ~~been successfully~~ stored in
the second lock object, ~~not accessing the data object.~~

8. (Currently Amended) A computer system for controlling access to a data object stored in a first storage location, the data object having an identifier (ID), the system comprising:

memory having program instructions;

storage means for storing data;

at least one processor to execute the program instructions to perform operations

a method comprising:

~~checking, before accessing the data object,~~ whether the ID is ~~contained~~ stored in a first lock object and whether the ID is associated with a second storage location; and

~~granting access to accessing the data object~~[[,]] if the ID is not ~~contained~~ stored in the first lock object ~~or if~~ and the ID is not yet associated with a second storage location;

~~determining whether a transactional lock has been successfully set on the data object;~~

~~determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and~~

~~granting read/write access to the data object based on the permanent lock not being set on the data object.~~

9. (Currently Amended) The computer system of claim [[13]] 8, wherein the method further comprising comprises:

deleting the ID from the ~~second~~ first lock object, if the ID is stored in the first lock object and the ID is not yet associated with a second storage location, after granting access to the data object.

10. (Currently Amended) The computer system of claim 8, wherein the first lock object comprises a table[[,]] having a first column for the ID and a second column for a link to the second storage location associated with the ID.

11. (Currently Amended) The computer system of claim 9, wherein the first lock object comprises a table[[,]] having a first column for the ID and a second column for a link to the second storage location associated with the ID.

12. (Original) The computer system of claim 8, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

13. (Currently Amended) The computer system of claim 8, wherein the method further comprises:

~~before performing the check,~~ storing the ID in a second lock object before checking whether the ID is stored in a first lock object and whether the ID is associated with a second storage location, ~~which is stored in a volatile storage means.~~

14. (Currently Amended) The computer system of claim 13, wherein the method further comprises:

checking[[,]] whether the ID ~~has been successfully~~ is stored in the second lock object before granting access to ~~accessing~~ the data object; and[[,]]

denying access to the data object if the ID has is not ~~been successfully~~ stored in the second lock object; ~~not accessing the data object.~~

15. (Currently Amended) A computer-readable medium comprising instructions for performing a method of controlling access to a data object stored in a first storage location, the data object having an identifier (ID), the medium method comprising ~~instructions for:~~

checking, ~~before accessing the data object,~~ whether the ID is ~~contained~~ stored in a first lock object and whether the ID is associated with a second storage location; and

granting access to ~~accessing~~ the data object[[,]] if the ID is not ~~contained~~ stored in the first lock object ~~or if~~ and the ID is not yet associated with a second storage location;—

~~determining whether a transactional lock has been successfully set on the data object;~~

~~determining whether a permanent lock has been set on the data object based on whether a transactional lock has been successfully set on the data object; and~~

~~granting read/write access to the data object based on the permanent lock not being set on the data object.~~

16. (Currently Amended) The medium of claim ~~[[20]]~~ 15, wherein the method further comprising comprises:

deleting the ID from the ~~second~~ first lock object, if the ID is stored in the first lock object and the ID is not yet associated with a second storage location, after granting access to the data object.

17. (Currently Amended) The medium of claim 15, wherein the first lock object comprises a table~~[[,]]~~ having a first column for the ID and a second column for a link to the second storage location associated with the ID.

18. (Currently Amended) The medium of claim 16, wherein the first lock object comprises a table~~[[,]]~~ having a first column for the ID and a second column for a link to the second storage location associated with the ID.

19. (Original) The medium of claim 15, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

20. (Currently Amended) The medium of claim 15, wherein the method further comprises:

~~before performing the check,~~ storing the ID in a second lock object before checking whether the ID is stored in a first lock object and whether the ID is associated with a second storage location, ~~which is stored in a volatile storage means.~~

21. (Currently Amended) The medium of claim 20, wherein the method
further comprises:

checking[[,]] whether the ID ~~has been successfully~~ is stored in the second lock
object before granting access to ~~accessing~~ the data object; and[[,]]

denying access to the data object if the ID ~~has~~ is not ~~been successfully~~ stored in
the second lock object; ~~not accessing the data object.~~

22. (Currently Amended) A memory for storing data for access by a process
being executed by a processor, the memory comprising:

a structure for controlling access to a data object stored in a first storage location,
the data object having an identifier (ID), the structure comprising:

a first lock object[[,]] storing the ID ~~of the data object~~ and a link, associated
with the ID, to a second storage location where the data object is stored;[[,]] and

a second lock object storing the ID of the data object; ~~and~~

instructions for:

~~determining whether a transactional lock has been successfully set on the~~
~~data object;~~

~~determining whether a permanent lock has been set on the data object~~
~~based on whether a transactional lock has been successfully set on the data object; and~~

~~granting read/write access to the data object based on the permanent lock~~
~~not being set on the data object.~~

23. (Currently Amended) The memory of claim 22, wherein the first lock object comprises a table[[,]] having a first column for the ID and a second column for the link to a storage location where the data object is stored.

24. (Original) The memory of claim 22, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

25. (Original) The memory of claim 23, wherein the data object comprises one or more fields of one or more tables and wherein the ID comprises one or more key fields of the one or more tables.

26. (Original) The memory of claim 22, wherein the first and second lock objects are created by a data moving or data archiving process.

27-28. (Canceled).